

Язык программирования PROLOG

Любая формула исчисления предикатов может быть представлена в виде конъюнкции дизъюнкций положительных или отрицательных литералов:

$$L_1 \vee L_2 \vee \dots \vee L_k,$$

где L_i – атомарная формула $F_i(t_1, t_2, \dots, t_m)$

t_i – терм

Метод доказательства: от противного: найти такие значения предметных переменных $X_1, X_2, \dots, X_n$, при которых формула истинна.

За основу декларативного описания предметной области принимается специальная форма представления в виде **фактов и правил**

Базовая формула (дизъюнкт Хорна)

$A \Leftarrow B$ - правило

Запрос (цель) – целевое утверждение, которое нужно доказать, исходя из множества фактов и правил программы

Процесс доказательства представляет собой выполнение программы

Для любых X, Y, Z если любит (X, Y) и нравится (Y, Z) , то нравится (X, Z)

нравится $(X, Z) \leftarrow$ любит (X, Y) & нравится (Y, Z)

Структура PROLOG-программы

ФАКТЫ

clauses

like (masha, flowers).

like (X,Z):- love (X,Y), like (Y,Z).

predicates

like (Name, Name)

love (Name, Name)

goal

like (masha, X).

Структура PROLOG-программы

domains

описание типов данных

predicates

описание предикатов

clauses

описание фактов и правил

Goal

цель

ПРАВИЛА:

отношение (объект, ...) if отношение (объект, ...)

Домены

symbol

integer

real

Goal

likes (X, reading).

domains

person, activity = symbol

age=integer

predicates

likes (person, activity)

clauses

likes (tom, basketball).

likes (mary, reading).

likes (elen, reading).

is_older (Person1, Person2) if
age (Person1, Age1) and
age (Person2, Age2) and
Age1>Age2.

Пример тезауруса на ПРОЛОГЕ

similar (big, gigantic).

similar (big, enomous).

similar (big, X).

similar (big, tall).

similar (big, huge).

similar (big, large).

Почти детективная история

мы знаем, что

человек (аллан, 25, м, футболист).

человек(аллан, 25, м, мясник).

человек (берт, 55, м, плотник).

человек(барбара, 22, ж, парикмахер).

человек(джон, 25, м, карманник).

знакомы(барбара, джон).

знакомы(барбара, берт).

знакомы(сюзанна, джон).

убит (сюзанна, клюшкой).

мотив(деньги).

мотив(ревность).

измазан (катрин, кровь).

измазан (аллан, грязь).

имеет (берт, дубинку).

имеет (джон, пистолет).

Из опыта мы знаем, что

одинаково_действует (дубинка, клюшка).

одинаково_действует (брусok, клюшка).

одинаково_действует (ножницы, нож).

одинаково_действует (бутса, клюшка).

действительно_имеет (Некто, бутсы) если

человек (Некто, __, __, футболист).

действительно_имеет (Некто, ножницы) если

человек (Некто, __, __, __). /* любой человек*/

**Подозреваем тех, кто имеет оружие, с помощью
которого убита Сюзанна**

подозревать (Некто) если

убита (сюзанна, Оружие) и

действует_одинаково (Объект, Оружие) и

действительно_имеет (Некто, Объект).

Подозреваем мужчин, знакомых с Сюзанной

подозревать (Некто) если

мотив (ревность) и

человек (Некто, __,м, __) и знакомы (сюзанна, Некто).

Женщина тоже могла быть убийцей

подозревать (Некто) если

мотив (ревность) и

человек (Некто, __,ж, __) и знакомы (Некто, Мужчина) и

знакомы (сюзанна, Мужчина).

Также можно подозревать и карманников...

Составные цели

domains

brand, color = symbol

age, price = integer

mileage = real

predicates

car (brand, mileage, age, color, price)

clauses

car(chrysler, 130000, 3, red, 12000).

car(ford, 90000, 4, gray, 25000).

car(mitsubishi, 20000, 1, black, 30000).

goal

car(Name,Odometer,Years_on_road,Body,Cost) and Cost <25000

Комментарии

/*

*/

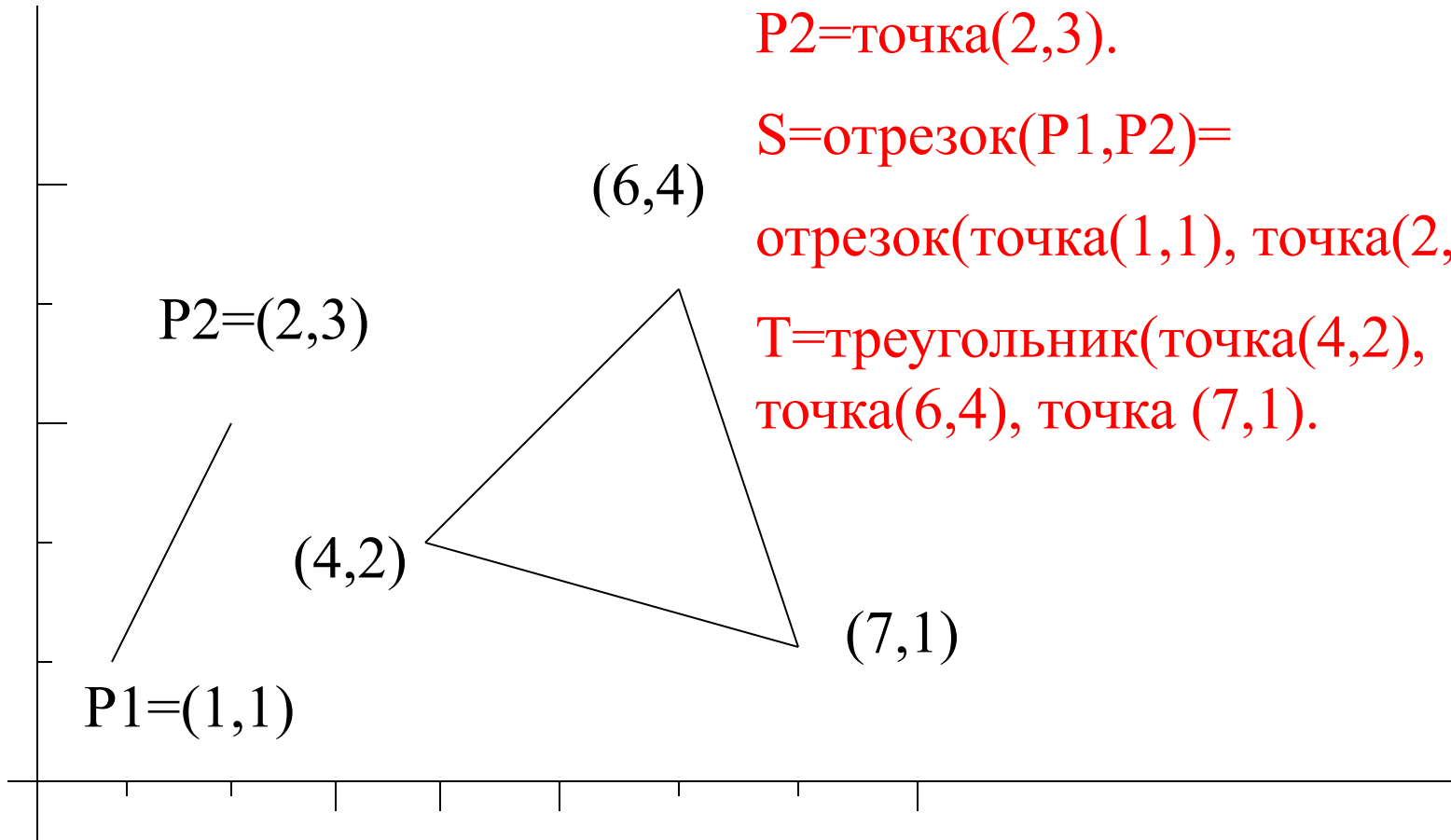
Некоторые примеры

$P1 = \text{точка}(1,1).$

$P2 = \text{точка}(2,3).$

$S = \text{отрезок}(P1, P2) =$
 $\text{отрезок}(\text{точка}(1,1), \text{точка}(2,3)).$

$T = \text{треугольник}(\text{точка}(4,2),$
 $\text{точка}(6,4), \text{точка}(7,1)).$



Обезьяна и банан

Есть комната. Возле двери стоит обезьяна.

В середине комнаты к потолку подвешен банан

Обезьяна хочет съесть банан, но достать до него она не может.

Около окна на полу есть ящик, который можно подвинуть

Возможные действия обезьяны:

Ходить по полу, залезать на ящик, двигать ящик, схватить банан

СМОЖЕТ ЛИ ОБЕЗЬЯНА СЪЕСТЬ БАНАН?

Будем считать, что обезьяний мир всегда находится в некотором состоянии, оно может изменяться со временем.

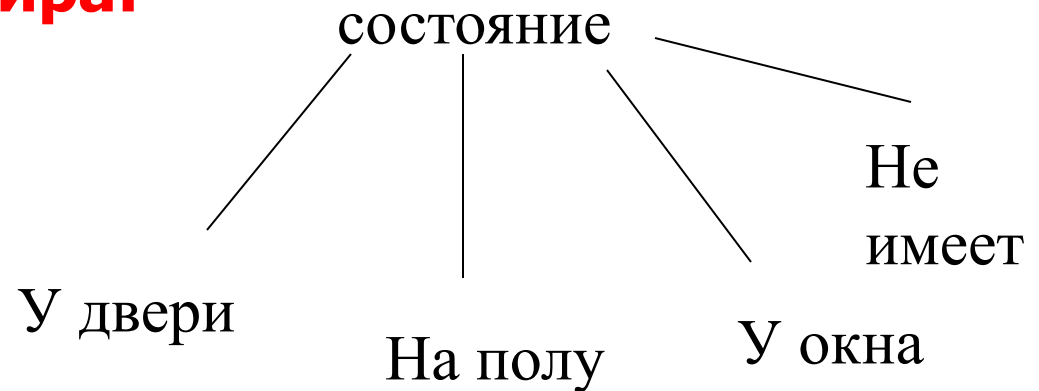
Исходное состояние мира:

Обезьяна у двери

Обезьяна на полу

Ящик у окна

Обезьяна не имеет банан



Цель игры: обезьяна имеет банан, т.е. любое состояние, где
состояние (_ _ _ имеет)

Типы ходов:

Не все ходы допустимы!

Схватить банан

Залезть на ящик

Подвинуть ящик

Перейти в другое место

ход (Состояние1, М, Состояние2)

Состояние1 -----> Состояние2

М

ход схватить: ход (состояние(середина, наящике, середина, неимеет),

схватить, состояние(середина, наящике, середина, имеет))

Обезьяна, находясь на полу, может перейти из любой горизонтальной позиции $P1$ в любую позицию $P2$.

Независимо от позиции ящика и независимо от наличия у нее банан

ход (состояние $(P1, \text{наполу}, B, H)$,
перейти $(P1, P2)$,
состояние $(P2, \text{наполу}, B, H)$).

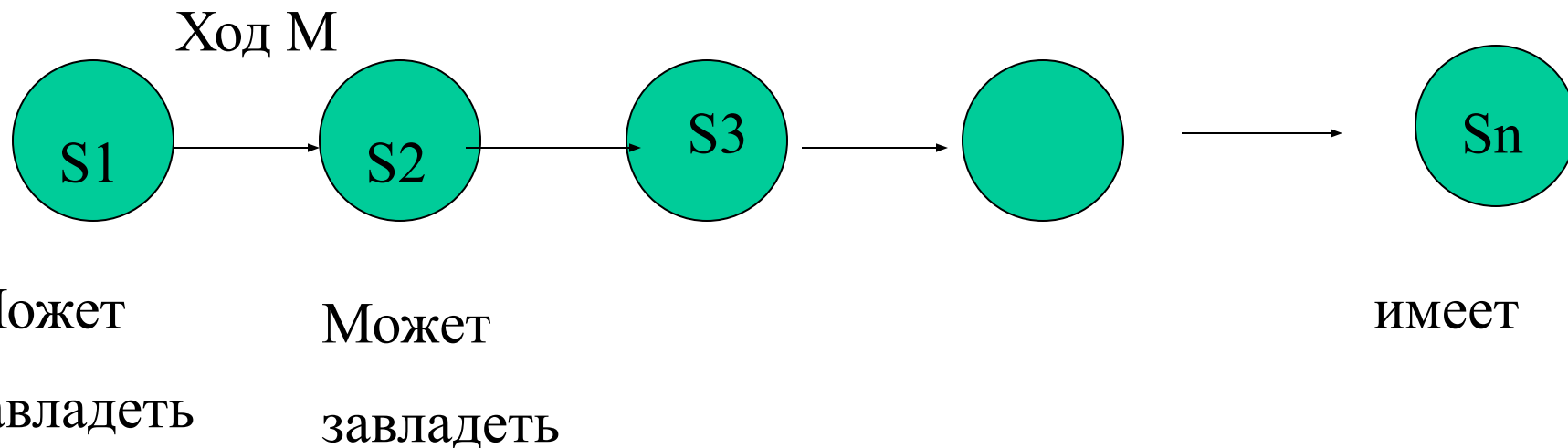
Некоторые утверждения:

Ход переводит из положения $P1$ в $P2$

Обезьяна на полу, как до, так и после хода

Ящик находится в некоторой точке B , которая не меняется после хода

Состояние «имеет банан» остается неизменным после хода



Главный вопрос: может ли обезьяна, находясь в некотором начальном состоянии S , завладеть бананом?

можетзавладеть(S)

Основные наблюдения обезьяны

1. Для любого состояния S , в котором обезьяна уже имеет банан, предикат **можетзавладеть** должен быть истинным. Ходов больше не требуется.

можетзавладеть (состояние($_$ $_$ $_$ имеет)).

2. Иначе может потребоваться один или несколько ходов.

Обезьяна может завладеть бананом в любом состоянии $S1$, если для него существует ход из состояния $S1$ в некоторое состояние $S2$, что попав в него, обезьяна уже может завладеть бананом.

**можетзавладеть ($S1$):- ход ($S1$, M , $S2$),
можетзавладеть ($S2$).**

Разрешенные ходы

ход (состояние(середина, наящике, середина, неимеет),
схватить, состояние(середина, наящике, середина, имеет)).

ход(состояние (P, наполу,P, H),
залесть, состояние (P, наящике, P, H).

ход (состояние (P1, наполу, P1, H),
подвинуть (P1, P2),
состояние (P2, наполу, P2, H)).

ход (состояние (P1, наполу, B, H),
перейти (P1, P2),
состояние (P2, наполу, B, H)).

можетзавладеть (состояние (__, __, __, имеет)).

можетзавладеть (Состояние1) : -

ход (Состояние1, Ход, Состояние2), можетзавладеть
(Состояние2).

Поиск банана обезьяной



Анонимные переменные

- В случае незначимости конкретного значения какой-либо переменной

predicates

car (brand,mileage,age, color, price)

Age=2, Cost=9000

clauses

car(ford, 10000, 1,black, 12000).

car (lada, 15000,2, white,9000).

goal

car (_, _, Age, _, Cost) and Cost <11000.

Порядок предложений и целей

Опасность бесконечного цикла

$r :- r.$ *R истинно, если r истинно.*

? – $r.$ *Бесконечный цикл*

В жизни обезьяны есть следующий порядок:

- Схватить
- Залезть
- Подвинуть
- перейти

У каждого есть рубашка
has (_, shirt).

Каждый умывается

washes (_).

domains

person=symbol

age=integer

predicates

student (person,age).

clauses

student(ivanov,17).

student(petrov,19).

student(sidorov,17).

BackTracing

Goal:

student(Man1,17) and
student(Man2,17)

and Man1 <>Man2

Man1=ivanov Man2=petrov

Man1=petrov Man2=ivanov

Использование NOT

marriage(sofie, X) :- male(X), not (smoker(X)).

Стандартные типы описаний

char символы в апострофах 'a'

integer целые числа -32768 до 32767

real числа с десятичной точкой или в экспоненциальной форме $-1e-307$ до $1e+308$

string строковый (последовательность символов, заключенная в кавычки)

symbol последовательность букв, чисел, знака подчеркивания, начинающаяся с маленькой буквы
если строка содержит пробелы "Mike likes tennis"

file

Многоуровневые объекты

book ("Tom Soyer", "Mark Twain").

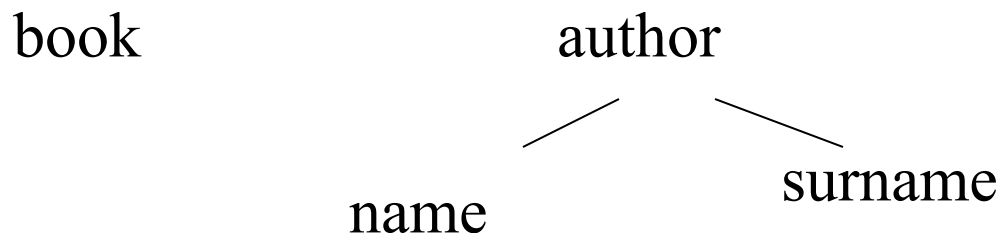
book ("Tom Soyer", author ("Mark", "Twain")).

Опишем автора как сложный объект

author = author (name, surname).

article= book (title, author).

title, name, surname=symbol



sentence = sentence (noun, verb).

noun = noun (word).

Рекурсия

- когда отношения описаны с помощью других отношений
- когда сложный объект является составной частью другого сложного объекта (рекурсивные объекты)

И еще раз про факториал!

domains

$n, f = \text{integer}$

predicates

$\text{factorial}(n, f).$

clauses

$\text{factorial}(1, 1).$

$\text{factorial}(N, \text{Res})$ if

$N > 1$ and

$N1 = N - 1$ and

$\text{factorial}(N1, \text{FacN1})$ and

$\text{Res} = N * \text{FacN1}.$

Для цели `factorial (2, Answer)`.

мы имеем

`factorial (2, Res) if 2>1, N1=N1=2-1, factorial (N1, FacN1),`

`Res=2 * FacN1.`

